

ОПЫТ СОЗДАНИЯ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ ГИБРИДНОЙ ЖИДКОСТНОЙ МОДЕЛИ

Д.А. Басавин¹, С.В. Поршневу²

¹ Белгород, УРФУ, basavind@gmail.com;

² Екатеринбург, УРФУ, sergey_porshnev@mail.ru

PARALLEL HYBRID FLUID MODEL

D.A. Basavin, S.V. Porshnev

Сегодня технологии параллельных вычислений на базе графических процессоров (Graphics Processor Unit – GPU) [1], несмотря на известные ограничения, существующие, в частности, при работе с памятью, позволяют решать множество сложных вычислительных задач, что исключает использование ресурсов дорогостоящих суперкомпьютеров.

Необходимость решения подобных задач существует, в том числе, и в сфере телекоммуникационных технологий, современное состояние которой характеризуется обусловлена стремительным ростом скоростей передачи данных (до десятков и сотен Гбит/с) и, соответствующим увеличением количества передаваемых в информационных каналах потоках. В этой ситуации, закономерно, что задача создания высокопроизводительных систем моделирования, предназначенных для поддержки принимаемых технических и конструкторских решений и имеющих относительно невысокую стоимость, является актуальной [2].

Следует отметить, что известные пакетные симуляторы трафика позволяют создавать математические модели только каналов передачи данных с умеренной пропускной способностью (до 10 Мбит/с). В этой связи для моделирования высокоскоростных Интернет-каналов (100 Мбит и более) в последние годы был предложен принципиально иной подход, в котором используется аналогия между потоком данных в Интернет-канале и течением жидкости [3]. Математическая модель, реализующая данный подход, представляет собой следующую систему дифференциальных уравнений (СДУ) с запаздыванием:

$$\frac{dW_i(t)}{dt} = \frac{I_{(W_i(t)-M_i)}}{R_i(t)} - \frac{W_i(t)}{2} \cdot \lambda_i(t), \quad (1)$$

$$\frac{dq_l(t)}{dt} = -I_{q(t)} \cdot C_l + \sum_{i \in N_l} A_i^l(t), \quad (2)$$

где W_i – текущий размер окна передачи данных передачи данных по протоколу TCP по i -му потоку²; q_l – длина очереди на входе l -го канала; $I_{f(t)}$ – функция Хевисайда:

$$I_{f(t)} = \begin{cases} 1, & \text{если } f(t) \geq 0, \\ 0, & \text{если } f(t) < 0; \end{cases}$$

$R_i(t)$ – время оборота i -го потока; $\lambda_i(t)$ – скорость потери пакетов i -го потока, C_l – пропускная способность l -го канала, который обслуживается данным маршрутизатором; $A_i^l(t)$ – скорость передачи i -го потока по l -му каналу:

$$A_i^l(t) = \frac{W_i^l(t)}{R_i^l(t)}.$$

² В рассматриваемой математической модели можно учесть механизмы и работы и других протоколов, используемых для передачи данных в сети Интернет, а также взаимное влияние информационных потоков, передаваемых по различным протоколам друг на друга. Однако это с неизбежностью приведет к существенному усложнению модели при относительно небольшом увеличении ее точности, так как Сети доминирует трафик, передаваемый по протоколу TCP. По литературным данным доля Интернет-трафика, передаваемого в соответствии с протоколом TCP, составляет свыше 90 % от общего объема информации, передаваемой в сети Интернет (см., например, Олифер В.Г., Олифер Н.А. Компьютерные сети. СПб.: Питер, 2007.)

Отметим, что как в базовой жидкостной модели Интернет-трафика, так и в ее известных модификациях, используется упрощенное описание источников трафика, как следствие, оценки его количественных характеристик оказываются весьма грубыми.

Для устранения отмеченного выше недостатка жидкостных моделей в [4] предложена гибридная жидкостная модель информационных потоков в магистральном Интернет-канале, представляющая собой комбинацию жидкостной модели [3] и статистического варианта модели абстрактных источников трафика. Гибридная модель позволяет описывать трафик в мультисервисных сетях с учетом присущего протоколу *TCP* механизма обратного влияния загрузки сети на работу источника, а также учитывать современные политики управления скоростью доступа к сети Интернет отдельных пользователей.



Рис. 1. Блок-схема алгоритм расчетов Интернет-трафика в соответствии с гибридной жидкостной моделью

В представленном выше алгоритме программы можно выделить следующие участки, распараллеливание которых представляется целесообразным:

1) генерация сценариев активности пользователей;

2) решение СДУ (1)–(2), так как все расчеты идентичны для каждого пользователя и могут производиться независимо друг от друга.

Отметим, что одним из наиболее важных вопросов, возникающих при реализации параллельных вычислений на графических процессорах, является выбор технологии параллельных вычислений (см., например, [5,6]), среди которых активно развиваются и используются в научных расчетах:

- Nvidia CUDA (технология параллельных вычислений, позволяющая разрабатывать масштабируемые (по аппаратным ресурсам) программы для GPU) [7];
- ATI DirectCompute [8] (технология поддерживается библиотекой OpenGL);
- OpenCL [9].

Каждая из технологий использует собственный подход к реализации как самих вычислений, так и реализации архитектуры приложений и вычислительной системы. Для реализации поставленной задачи можно использовать любую из перечисленных выше, однако технологии Nvidia CUDA и ATI DirectCompute могут использоваться только на видеокартах и графических ускорителях одноименных компаний (Nvidia и ATI), что программные параллельные реализации гибридной жидкостной модели универсальности. В свою очередь технология OpenCL разрабатывается как единый стандарт [9] для параллельных вычислений, позволяющий производить вычисления на любой платформе, поддерживающей данный стандарт. Данный стандарт поддерживает реализацию технологии как на видеокартах и графических ускорителях, так и на многоядерных процессорах, много процессорных системах и вычислительных кластерах. Таким образом, код, написанный в соответствии со стандартом OpenCL, может быть запущен как на видеокартах, так и графических ускорителях компаний Nvidia, ATI и процессорах AMD, Intel.

В настоящее время авторами разработана пилотная версия параллельной программной реализации гибридной жидкостной модели, которая находится в стадии тестовых испытаний. Их результаты свидетельствуют о ее работоспособности. Однако сравнение скорости ее работы с аналогичной реализацией, использующей вычислительные возможности CPU, не обнаруживают существенного увеличения скорости вычислений, что свидетельствует о необходимости оптимизации кода. Проведенный анализ результатов тестовых испытаний позволил выявить следующие основные причины недостаточно высокой скорости вычислений пилотной версии параллельной программной реализации гибридной жидкостной модели:

- 1) недостаточно эффективная организация размещения данных в памяти GPU;
- 2) «непараллельность» алгоритма метода Рунге-Кутты, используемого для решения СДУ.

Для устранения выявленных недостатков авторы планируют оптимизировать распределение данных в памяти GPU, а также использовать параллельные алгоритмы численного решения СДУ (см., например, [10])

Литература

1. Херн Д.М., Бейкер П. Компьютерная графика и стандарт OpenGL = Computer Graphics with OpenGL. – М.: Вильямс, 2005. – 1168 с.
2. Hernandez-Campos F. Generating realistic TCP workloads / F. Hernandez-Campos, F.D. Smith, and K. Jeray // In Proceedings of Computer Measurement Group (CMG) Conference, 2004. P. 273–284.
3. Liu Y. Fluid Models and Solutions for Large-Scale IP Networks / Y. Liu, F. L. Presti, V. Misra, D. Towsley, Y. Gu. // Performance Evaluation Review, ACM SIGMETRICS, Volume 31, Issue 1, P. 91–101, 2003.
4. Гребенкин М.К., Поршнева С.В. Гибридная жидкостная модель магистрального Интернет-канала / М.К. Гребенкин, С.В. Поршнева// Saarbrücken: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. 163 с.
5. Боярченко А.С., Поташиков С.И. Использование графических процессоров и технологии CUDA для задач молекулярной динамики / А.С. Боярченко, С.И. Поташиков// Вычисление и программирование, 2009. –Т. 10. – С. 9–23.
6. Боярченко А.С., Поташиков С.И. Параллельная молекулярная динамика с суммированием Эвальда и интегрированием на графических процессорах / А.С. Боярченко, С.И. Поташиков// Вычисление и программирование, 2009. –Т. 10. – С. 158–175.
7. http://en.wikipedia.org/wiki/Cuda#Supported_GPUs
8. http://en.wikipedia.org/wiki/Comparison_of_ATI_Graphics_Processing_Units
9. Khronos OpenCL Working Group, The OpenCL Specification [Электронный ресурс]. Режим доступа: <http://www.khronos.org/registry/cl/specs/opencl-1.0.pdf>
10. Pierluigi A., Luigi B. Recent Advances in the Parallel Solution in Time of ODEs // NUMERICAL ANALYSIS AND APPLIED MATHEMATICS: International Conference on Numerical Analysis and Applied Mathematics, 2008. AIP Conference Proceedings, Volume 1048, pp. 867–870 (2008).